Leo R. Beus *(pro hac vice)*
Michael K. Kelly *(pro hac vice)*
BEUS GILBERT MCGRODER PLLC
Attorneys at Law
701 North 44th Street
Phoenix, Arizona 85008-6504
Telephone: (480) 429-3000 | Facsimile: (480) 429-3001
Email: lbeus@beusgilbert.com | mkelly@beusgilbert.com

Allan Steyer (State Bar No. 100318)
Suneel Jain (State Bar No. 314558)
STEYER LOWENTHAL BOODROOKAS
ALVAREZ & SMITH LLP
235 Pine Street, 15th Floor
San Francisco, California 94104
Telephone: (415) 421-3400 | Facsimile: (415) 421-2234
E-mail: asteyer@steyerlaw.com | sjain@steyerlaw.com

Attorneys for Defendant and Counterclaimant
Swarm Technology LLC

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

JUNIPER NETWORKS, INC. and APSTRA, INC., Plaintiffs, v. SWARM TECHNOLOGY LLC, Defendant.	Case No. 5:20-cv-03137-JD **SWARM TECHNOLOGY LLC'S ANSWER TO FIRST AMENDED COMPLAINT AND COUNTERCLAIMS FOR PATENT INFRINGEMENT AGAINST JUNIPER NETWORKS, INC. AND APSTRA, INC.** **DEMAND FOR JURY TRIAL**
SWARM TECHNOLOGY LLC, Counterclaimant, v. JUNIPER NETWORKS, INC. and APSTRA, INC., Counterdefendant.	

Defendant/Counterclaimant Swarm Technology LLC, an Arizona limited liability company ("Swarm"), hereby Answers the First Amended Complaint for Declaratory Judgment of Non-Infringement filed 2 April 2021 by Juniper Networks, Inc. ("Juniper") and Apstra, Inc. ("Apstra"), a wholly owned subsidiary of Juniper (collectively, "Plaintiffs/Counterdefendants"), and counterclaims against Plaintiffs for patent infringement under Title 35 of the United States Code. Swarm alleges the following upon personal knowledge and belief as where appropriate, and otherwise upon information and belief:

ANSWER

1. Swarm ADMITS the allegations contained in Paragraph 1 of the Amended Complaint.

2. Swarm ADMITS the allegations contained in Paragraph 2 of the Amended Complaint.

3. Swarm ADMITS the allegations contained in Paragraph 3 of the Amended Complaint.

4. Swarm DENIES the allegations contained in Paragraph 4 of the Amended Complaint.

5. Swarm ADMITS that Swarm offered a license to Apstra, and DENIES the remaining allegations contained in Paragraph 5 of the Amended Complaint.

6. Swarm ADMITS the allegations contained in Paragraph 6 of the Amended Complaint.

7. Swarm ADMITS the allegations contained in Paragraph 7 of the Amended Complaint.

8. Swarm ADMITS the allegations contained in Paragraph 8 of the Amended Complaint.

9. Swarm ADMITS that Swarm is an Arizona limited liability company having a principal place of business at 732 East Lehi Road, Mesa, Arizona 85203, and denies the remaining allegations contained in Paragraph 9 of the Amended Complaint.

10. Swarm ADMITS the allegations contained in Paragraph 10 of the Amended Complaint.

11. This Court has found personal jurisdiction over Swarm in this District, and Swarm DENIES the remaining allegations contained in Paragraph 11.

12. Swarm ADMITS that this Court has found personal jurisdiction over Swarm in this District, and DENIES the remaining allegations contained in Paragraph 12.

13. Swarm ADMITS that it communicated with other companies in this District regarding the Patents-in-Suit, and DENIES the remaining allegations contained in Paragraph 13 of the Amended Complaint.

14. Swarm ADMITS that it communicated with Cisco Systems, Inc., and Arista Networks, Inc., regarding the Patents-in-Suit, and DENIES the remaining allegations contained in Paragraph 14 of the Amended Complaint.

15. Swarm ADMITS the allegations contained in Paragraph 15 of the Amended Complaint.

16. Swarm ADMITS the allegations contained in Paragraph 16 of the Amended Complaint.

17. Swarm ADMITS the allegations contained in Paragraph 17 of the Amended Complaint.

18. Swarm DENIES the allegations contained in Paragraph 18 of the Amended Complaint.

19. This Court has found venue proper in this District.

20. Swarm ADMITS the allegations contained in Paragraph 20 of the Amended Complaint.

21. Swarm ADMITS the allegations contained in Paragraph 21 of the Amended Complaint.

22. Swarm ADMITS the allegations contained in Paragraph 22 of the Amended Complaint.

23. Swarm ADMITS the allegations contained in Paragraph 23 of the Amended Complaint.

24. Swarm ADMITS the allegations contained in Paragraph 24 of the Amended Complaint.

25. Swarm ADMITS the allegations contained in Paragraph 25 of the Amended Complaint.

26. Swarm ADMITS that the listed Juniper product numbers appear on the ’004 Patent claim chart sent from Swarm to Juniper, and Denies the remaining allegations of Paragraph 26.

27. Swarm ADMITS the allegations contained in Paragraph 27 of the Amended Complaint.

28. Swarm ADMITS the allegations contained in Paragraph 28 of the Amended Complaint.

29. Swarm objects to the use of the word “warned,” and otherwise ADMITS the remaining allegations contained in Paragraph 29 of the Amended Complaint.

30. Swarm ADMITS the allegations contained in Paragraph 30 of the Amended Complaint.

31. Swarm ADMITS the allegations contained in Paragraph 31 of the Amended Complaint.

32. Swarm ADMITS that the quoted language appears in Swarm’s November 26, 2019 email, and DENIES the remaining allegations contained in Paragraph 32 of the Amended Complaint.

33. Swarm ADMITS the allegations contained in Paragraph 33 of the Amended Complaint.

34. Swarm ADMITS the allegations contained in Paragraph 34 of the Amended Complaint.

35. Swarm ADMITS the allegations contained in Paragraph 35 of the Amended Complaint.

36. Swarm ADMITS the allegations contained in Paragraph 36 of the Amended Complaint.

37. Swarm ADMITS the allegations contained in Paragraph 37 of the Amended Complaint.

38. Swarm ADMITS the allegations contained in Paragraph 38 of the Amended Complaint.

39. Swarm ADMITS the allegations contained in Paragraph 39 of the Amended Complaint.

40. Swarm ADMITS the allegations contained in Paragraph 40 of the Amended Complaint.

41. Swarm DENIES the allegations contained in Paragraph 41 of the Amended Complaint.

42. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 42 of the Amended Complaint, and therefore DENIES the same.

43. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 43 of the Amended Complaint, and therefore DENIES the same.

44. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 44 of the Amended Complaint, and therefore DENIES the same.

45. Swarm DENIES the allegations contained in Paragraph 45 of the Amended Complaint.

46. Swarm ADMITS that the ’777 Patent describes embodiments in which a solidarity cell retrieves tasks without requiring an instruction from the CPU, and DENIES the remaining allegations contained in Paragraph 46 of the Amended Complaint.

47. Swarm DENIES the allegations contained in Paragraph 47 of the Amended Complaint.

48. Swarm DENIES the allegations contained in Paragraph 48 of the Amended Complaint.

49. Swarm DENIES the allegations contained in Paragraph 49 of the Amended Complaint.

50. Swarm DENIES the allegations contained in Paragraph 50 of the Amended Complaint.

51. Swarm ADMITS the allegations contained in Paragraph 51 of the Amended Complaint.

52. Swarm DENIES the allegations contained in Paragraph 52 of the Amended Complaint.

53. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 53 of the Amended Complaint, and therefore DENIES the same.

54. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 54 of the Amended Complaint, and therefore DENIES the same.

55. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 55 of the Amended Complaint, and therefore DENIES the same.

56. Swarm DENIES the allegations contained in Paragraph 56 of the Amended Complaint.

57. Swarm ADMITS that the ’004 Patent describes embodiments in which a co-processor retrieves tasks from a task pool without any communication between the first co-processor and the controller, and DENIES the remaining allegations contained in Paragraph 57.

58. Swarm DENIES the allegations contained in Paragraph 58 of the Amended Complaint.

59. Swarm DENIES the allegations contained in Paragraph 59 of the Amended Complaint.

60. Swarm DENIES the allegations contained in Paragraph 60 of the Amended Complaint.

61. Swarm DENIES the allegations contained in Paragraph 61 of the Amended Complaint.

62. Swarm ADMITS the allegations contained in Paragraph 62 of the Amended Complaint.

63. Swarm DENIES the allegations contained in Paragraph 63 of the Amended Complaint.

64. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 64 of the Amended Complaint, and therefore DENIES the same.

65. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 65 of the Amended Complaint, and therefore DENIES the same.

66. Swarm lacks sufficient knowledge as to the factual allegations contained in Paragraph 66 of the Amended Complaint, and therefore DENIES the same.

67. Swarm DENIES the allegations contained in Paragraph 67 of the Amended Complaint.

68. Swarm ADMITS that the ’275 Patent describes embodiments in which a co-processor retrieves tasks from a task pool without any communication between the first co-processor and the controller, and DENIES the remaining allegations contained in Paragraph 68.

69. Swarm DENIES the allegations contained in Paragraph 69 of the Amended Complaint.

70. Swarm ADMITS that the '275 Patent describes embodiments in which a co-processor retrieves tasks from a task pool without any communication between the first co-processor and the controller, and DENIES the remaining allegations contained in Paragraph 70.

71. Swarm DENIES the allegations contained in Paragraph 71 of the Amended Complaint.

72. Swarm DENIES the allegations contained in Paragraph 72 of the Amended Complaint.

COUNTERCLAIMS FOR PATENT INFRINGEMENT

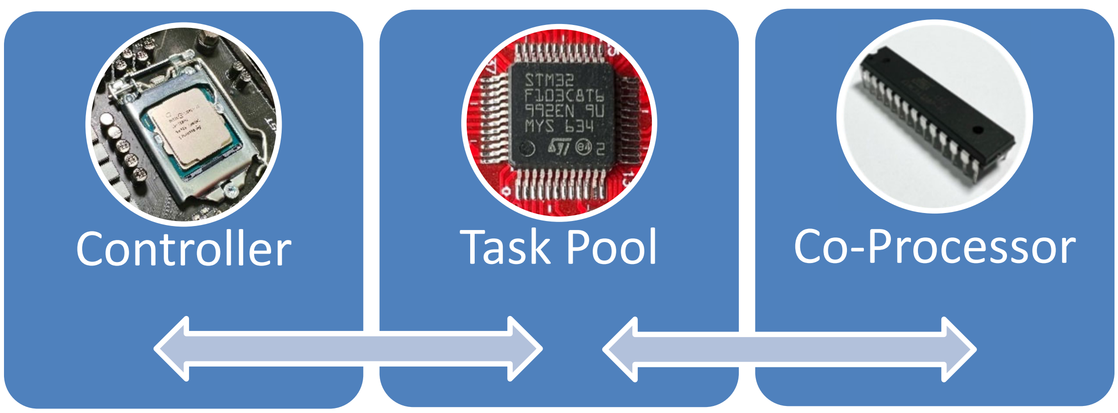
BACKGROUND

73. Alfonso Íñiguez is the sole inventor of three (3) United States Patents, a Japanese patent, and has a number of additional patent applications currently pending in the United States, the European Union, India, China, and Hong Kong relating to a revolutionary new computer processing architecture.

74. Prior to Mr. Íñiguez' invention, conventional multiprocessors included a central processing unit ("CPU") and one or more co-processors (see illustration below). That is, the processing "architecture" consisted of a primary controller which distributed tasks directly to a plurality of co-processors. This conventional approach is disadvantageous in that a significant amount of the CPU's processing cycles (bandwidth) is consumed by task distribution, while the co-processors often remain idle while waiting for a new task from the CPU.

Conventional Architecture

75. Mr. Íñiguez modified the arrangement of components (the architecture) of multiprocessing systems (see illustration below) by interposing an intermediate device – the task pool – between the CPU and the co-processors. In addition, Mr. Íñiguez modified the way each of those components operate, both individually and in combination with each other. As a result, Mr. Íñiguez invented a new multiprocessor system architecture which had never existed before.



Swarm Architecture

76. The United States Patent and Trademark Office has awarded, *inter alia*, the following Patents to Mr. Íñiguez: i) U.S. Patent No. 9,852,004 issued December 26, 2017, and entitled "System and Method for Parallel Processing Using Dynamically Configurable Proactive Co-Processing Cells" (the '004 Patent); ii) U.S. Patent No. 10,592,275 issued March 17, 2020, and entitled "System and Method for Swarm Collaborative Intelligence Using Dynamically Configurable Proactive Autonomous Agents" (the '275 Patent); and iii) U.S. Patent No. 9,146,777 issued September 29, 2015, and entitled "Parallel Processing With Solidarity Cells By Proactively Retrieving From a Task Pool a Matching Task for the Solidarity Cell to Process" (the '777 Patent). True and correct copies of the '004 Patent, the '275 Patent, and the '777 Patent (collectively, the "Patents-in-Suit") are attached as Exhibits A, B, and C, respectively, and incorporated herein by this reference. As set forth

more fully below and in the claim charts attached as Exhibits D, E, and F (the "Claim Charts"), Plaintiffs infringe all of Claims 1-3 and 7-12 of the '004 Patent, Claims 1-4, 6-7, and 9-17 of the '275 Patent, and Claims 1-10 and 12-14 of the '777 Patent, either directly, contributorily, or through inducement (35 U.S.C. § 271). Additional references, including literature describing Plaintiffs' products and services, are cited in the Claim Charts and are attached hereto as Exhibits G-Q.

77. Alfonso and Alejandra Íñiguez founded Swarm Technology, LLC as an Arizona Limited Liability Company on January 17, 2014. Pursuant to written assignments from Mr. Íñiguez, the Patents-in-Suit are now owned by Swarm Technology, LLC.

78. In recent years the cloud computing industry, led by Plaintiffs Juniper and Apstra, has migrated away from the traditional "primary/secondary" model – in which a central controller directly controls a plurality of microprocessors – to a distributed "co-processing" model. The new co-processing model does not require direct communication between the controller and the co-processors. Instead, coordination between the controller (typically a desktop, laptop, or hand-held computer) and the co-processors involves an intermediary data structure referred to as a "task pool." A credentialed administrator uses the controller to populate the task pool with discrete tasks to be performed by the co-processors. Each co-processor proactively retrieves tasks directly from the task pool and notifies the task pool when each task is completed. This allows the controller to indirectly accomplish multiple tasks without having to expend unnecessary processing cycles directly supervising the co-processors.

79. As detailed in the attached Claim Charts, the systems and methods used in Plaintiffs' cloud computing products and services are precisely the same as those claimed in the Patents-in-Suit. Consequently, Juniper and Apstra are jointly and severally liable to Swarm for infringing the Patents-in-Suit.

THE PARTIES

80. Swarm Technology, LLC, is an Arizona limited liability company (Arizona Entity ID L18990310) with its principal place of business at 732 East Lehi Road, Mesa, Arizona 85203.

81. Alfonso Íñiguez is the inventor of the Patents-in-Suit, a Member of Swarm Technology, LLC, and a resident of Mesa, Arizona.

82. Alejandra Íñiguez is a Member of Swarm Technology, LLC, and a resident of Mesa, Arizona.

83. Alfonso and Alejandra Íñiguez are husband and wife and are the sole owners of Swarm Technology, LLC.

84. Plaintiff Juniper is a Delaware corporation having its principal place of business at 1133 Innovation Way, Sunnyvale, California 94089.

85. Apstra is a Delaware corporation having its principal place of business at 333 Middlefield Road, Suite 200, Menlo Park, CA 94025.

SUBJECT MATTER JURISDICTION

86. This action arises under the Patent Act of the United States of America, 35 U.S.C. § 1, *et seq*.

87. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

PERSONAL JURISDICTION AND VENUE

88. 35 U.S.C. § 271 provides, in pertinent part:

> (a) Except as otherwise provided in this title, whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent.

> (b) Whoever actively induces infringement of a patent shall be liable as an infringer.
>
> (c) Whoever offers to sell or sells within the United States or imports into the United States a component of a patented machine, manufacture, combination or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the invention, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial noninfringing use, shall be liable as a contributory infringer.

89. Plaintiffs have offered and continue to offer infringing products and services in this judicial District.

90. Both Plaintiffs have conceded that personal jurisdiction and venue are proper in this District. (Document 38).

<u>THE STORY BEHIND MR. ÍÑIGUEZ’ INVENTIONS</u>

91. Alfonso Íñiguez was born in Tijuana, Mexico in 1965. He is pictured below (on the far right) with his mother and three siblings in approximately 1970:

92. Alfonso displayed remarkable abilities in science, technology, and mathematics at an early age. He also had an interest in how colonies of ants communicate with one another to perform tasks for the benefit of the queen, but without communicating with the queen directly. He would later apply his study of ant colonies to his design and development of co-processing architectures for computer systems.

93. While working at the American Consulate in Nogales, Mexico, Alfonso's mother obtained a United States Green Card. After leaving her employment at the Consulate in 1975, she submitted a Green Card application for Alfonso when he was ten (10) years old. Instilled with an impeccable work ethic, Alfonso went on to receive a Bachelor of Science degree in Computer Engineering from the Universidad Autonoma de Guadalajara, México in 1989.

94. Alfonso obtained his Green Card in 1987 and emigrated to the United States in 1989 to pursue graduate studies. While working full-time in various computer-related fields, Mr. Íñiguez attended the University of Arizona in Tucson, Arizona, and became a U.S. Citizen in 1994. In 1995, he was awarded a Master of Science degree in Electrical Engineering from the University of Arizona.

95. During the 2009 recession, Mr. Íñiguez was one of many employees laid off at Freescale Semiconductor (formerly Motorola, Inc.). After an extensive search, he secured an interview with a leading chip manufacturer as a Computer Architect.

96. Mr. Íñiguez prepared for his interview by reading books, papers, and performing extensive research in the field of computer architecture. He was struck by the inefficiencies associated with state-of-the-art computer processing architectures. He intuitively knew there was a better way for computer processors to cooperate with each other and with a central controller to perform complex processing tasks. He also believed that the swarm intelligence exhibited by ant colonies would play an important role in his new processing paradigm.

97. Drawing on his computer industry experience, Mr. Íñiguez identified two major drawbacks with existing multiprocessing frameworks. First, a significant portion of the CPU's processing cycles (bandwidth) was consumed assigning tasks to the co-processors. Second, the processors were often idle while waiting for a new task.

98. To address these shortcomings, Mr. Íñiguez invented a revolutionary new parallel processing paradigm, generally characterized by co-processors configured to proactively seek new tasks without having to communicate directly with (or wait for) the CPU. These co-processors include hardware and/or software components which are variously referred to as "autonomous agents" configured to retrieve "tasks"."

99. On January 25, 2013, Mr. Iñiguez filed his first utility patent application with the United States Patent and Trademark Office, and thereafter filed additional utility patent applications, each claiming priority to the original January 2013 filing date.

100. On September 29, 2015, the United States Patent and Trademark Office (the "USPTO") awarded U.S. Patent No. 9,146,777 entitled "Parallel Processing with Solidarity Cells by Proactively Retrieving from a Task Pool a Matching Task for the Solidarity Cell to Process" to Swarm.

101. On 26 December 2017, the USPTO awarded U.S. Patent No. 9,852,004 entitled "System and Method for Parallel Processing using Dynamically Configurable Proactive Co-Processing Cells" to Swarm.

102. On 17 March 2020, the USPTO awarded U.S. Patent No. 10,592,275 entitled "System and Method for Swarm Collaborative Intelligence using Dynamically Configurable Proactive Autonomous Agents" to Swarm.

103. Swarm is the sole owner of all right, title, and interest in and to each of the foregoing Patents-in-Suit.

104. Various products and services made, used, sold, offered for sale, or imported into the Unites States by Plaintiffs embody every element of at least one claim of each of the Patents-in-Suit, whether directly, contributorily, and/or through inducement (35 U.S.C. § 271), either literally or under the doctrine of equivalents.

105. The Patents-in-Suit disclose several embodiments, including a processing system having a controller configured to populate a task pool and one or more co-processors configured to proactively retrieve tasks from the task pool. In this way, the controller communicates directly with the task pool, and indirectly with the co-processors through the task pool.

106. Mr. Iñiguez contemplated many practical applications of his inventions, one of which included networks comprising Internet of Things (IoT) networks and supporting devices. One problem faced by engineers and computer architects surrounds the control of large numbers of devices linked to an IoT network, and how to harness their collective processing capacity without over-burdening the CPU.

107. The demand for IoT devices and IoT networks continues to drive growth in cloud-based products and services involving computing, storage, networking, databases, analytics, application services, deployment, mobile tools, and developer tools. Present day IoT networks make these services available to virtually any device connected to the Internet.

108. To help explain the technology, Mr. Íñiguez and his family created a series of homemade videos, each lasting approximately four minutes. The videos feature teams of ants working together and completing tasks, giving a visual metaphor of his revolutionary new computer architecture. These videos may be viewed at:

https://vimeo.com/150745850 (CPU Architecture - Prior Art)

https://vimeo.com/150748582 (Co-Processor Architecture - Prior Art)

https://vimeo.com/150450660 (Solidarity Cell Architecture - Issued Patent)

https://vimeo.com/150743111 (Heterogeneous Parallel Processing - Issued Patent)

https://vimeo.com/150759740 (Internet of Things Parallel Processing - Issued Patent)

https://vimeo.com/150744874 (Robot Automation - Issued Patent)

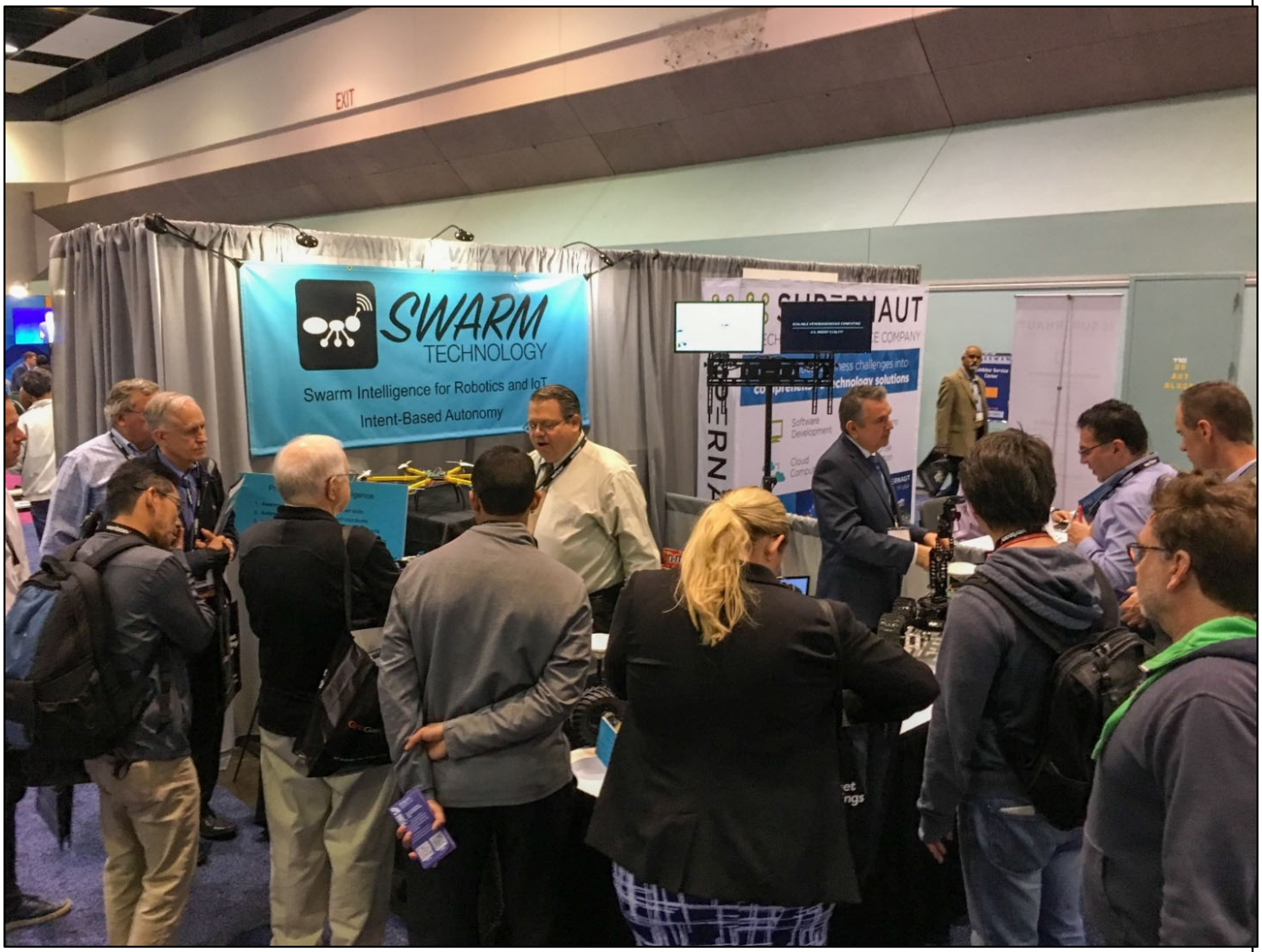
https://vimeo.com/177881911 (Ants, Robots, IoT and Parallel Processing)

109. Mr. Íñiguez and his family presented his technology at trade shows and other industry events, such as the: i) "Internet of Things World Conference 2018," Santa Barbara California, May 14 – 17, 2018; ii) "IoT Tech Expo North America 2017," Santa Clara, California, November 29-30, 2017; iii) "International Conference on Intelligent Robots and Systems (IROS) 2017," Vancouver, Canada, September 24–28, 2017; and iv) "Internet of Things World Conference 2017," Santa Clara, California, May 16-18, 2017.

110. Below is a photograph (left-to-right) of the Íñiguez family including sons Ulises and Isaac, daughter Daniela, wife Alejandra, and husband Alfonso promoting Swarm at an industry event in 2017:

111. Ulises Íñiguez recently graduated from the University of Notre Dame with a B.S. in Mechanical Engineering. Isaac Íñiguez is currently a Senior at Franciscan University studying Business and Marketing. Daniela Íñiguez is currently a Sophomore at Franciscan University studying Engineering (scheduled to transfer to the University of Notre Dame to complete a B.S. in Mechanical Engineering). All three Íñiguez children have studied on academic scholarships; currently, Isaac and Daniela are studying on academic scholarships.

112. Below is a photograph of Alfonso Íñiguez (right) and his cousin Pablo Garcia (B.S. Industrial Engineering - Instituto Tecnológico de Sonora, Mexico) promoting Swarm at an industry event in 2018:



113. Mr. Íñiguez's technology has also been the subject of news articles and other press coverage, such as the IEEE News in May of 2017, the Business News in April of 2018, the East Valley Tribune in April 2016, the Business Journal in December of 2015, and the EE Times in December of 2017, among others.

114. Mr. Íñiguez is also the author of a peer reviewed research paper published by the International Conference on Agents and Artificial Intelligence held in Porto, Portugal, in 2017. The International Conference on Agents and Artificial Intelligence is the most prestigious Artificial Intelligence conference in the World. It is extremely rare to include a company researcher (as opposed to a university researcher) as a featured author.

115. Around 2014, Mr. Íñiguez began to discover that many technology companies have incorporated his technology into their own products and services and are

marketing them to their customers. Mr. Íñiguez determined that at least the following product and services promoted by Plaintiffs infringe the Patents-in-Suit: i) Juniper Apstra 4.0. Product literature promoting and offering these services for sale in Arizona may be viewed at: https://www.juniper.net/us/en/products/network-automation/apstra/apstra-system.html.

116. After Mr. Íñiguez's first patent issued in September 2015, Swarm began offering Juniper and Apstra patent licensing opportunities.

117. The Plaintiffs were provided written notice of the Patents-in-Suit several times between 2015 and 2021. As such, their infringement may be willful under 35 U.S.C. § 284.

118. The '004 Patent describes a system and method for parallel processing using dynamically configurable proactive co-processing cells.

119. One embodiment described in the '004 Patent includes a controller configured to populate a task pool. One or more co-processors, or solidarity cells, are configured to proactively retrieve tasks from the task pool. Each co-processor includes an agent that interrogates the task pool to seek a task to perform. As a result, the co-processors work together "in solidarity" with each other and with the task pool. Each co-processor is autonomous in that it may interact with the task pool without being instructed to do so by the controller. Devices and their associated co-processors may be added to a network on a "plug and play" basis.

120. A representative figure is below:

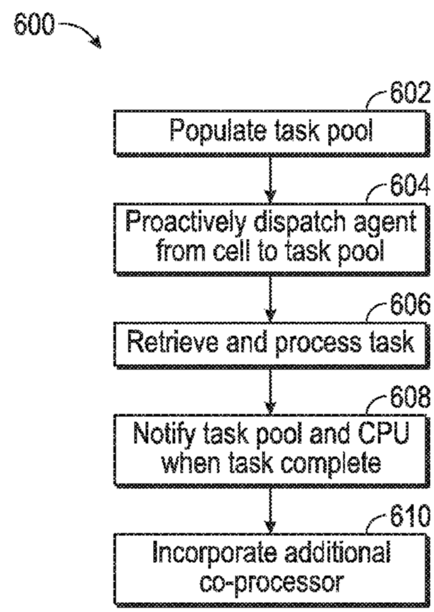


FIG. 6

121. The controller populates the task pool (602). The co-processors proactively dispatch an agent to the task pool (604). Each co-processor retrieves and processes a task (606). The task pool and controller are notified when the task is complete (608). An additional co-processor may be incorporated, "on boarded," or provisioned (610).

122. Claim 1 of the '004 Patent relates to a processing system including a task pool, a controller, and first and a second co-processors each configured to retrieve tasks from the task pool and update the task pool once the task is processed, without requiring direct communication with the controller.

123. Claim 1 of the '004 Patent is set forth in its entirety below:

1. A processing system, comprising:

a task pool;

a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;

> a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller; and
>
> a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second co-processor; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller;
>
> wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller.

124. As discussed below in conjunction with publicly available literature, many of the Plaintiffs' products and services embody all of the foregoing elements of Claim 1, as well as claims 2-3 and 7-12 of the '004 Patent.

125. As a result of Plaintiffs' infringement of the '004 Patent, Swarm has incurred substantial monetary and other damages.

126. The '275 Patent describes a system and method for collaborative intelligence using dynamically configurable proactive autonomous agents.

127. Claim 1 of the '275 Patent relates to a collaborative intelligence system including a task pool, a controller configured to populate the task pool with a plurality of tasks, and first and second co-processors each configured to retrieve tasks from the task pool and update the task pool to reflect completion of the task, without requiring direct communication with the controller, and to autonomously work together in solidarity with the task pool to complete a common objective.

128. Claim 1 of the '275 Patent is set forth in its entirety below

> 1. A collaborative intelligence system, comprising:

> a task pool;
>
> a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;
>
> a first co-processor configured to successively: proactively retrieve a first task from the task pool; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller; and
>
> a second co-processor configured to successively: proactively retrieve a second task from the task pool; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller;
>
> wherein the collaborative intelligence system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller;
>
> the plurality of first tasks and the plurality of second tasks are associated with a common objective;
>
> the first and second co-processors autonomously work together in solidarity with the task pool to complete the common objective.

129. As detailed below in conjunction with publicly available literature many of the Plaintiffs' products and services embody all of the foregoing elements of Claim 1, as well as claims 2-4, 6-7, and 9-17 of the '275 Patent.

130. As a result of Plaintiffs' infringement of the '275 Patent, Swarm has incurred substantial monetary and other damages.

131. The '777 Patent describes an apparatus for parallel processing of a large computing requirement.

132. Claim 1 of the ’777 Patent includes a CPU, a task pool, and a solidarity cell having an agent configured to proactively retrieve a matching task from the task pool, without requiring an instruction from the CPU.

133. Claim 1 of the ’777 Patent is set forth in its entirety below:

> 1. An apparatus for parallel processing of a large computing requirement, the apparatus comprising:
>
> a central processing unit (“CPU”);
>
> a task pool in electronic communication with the CPU; and
>
> a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process;
>
> wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks;
>
> wherein each task comprises a descriptor, the descriptor containing at least:
>
> > a function to be executed; and
> >
> > a memory location of data upon which the function is to be executed;
> >
> > wherein the first agent is a data frame comprising:
> >
> > a source address, a destination address and a payload;
>
> wherein the first agent retrieves the matching task by:
>
> > being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform;
> >
> > searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and

> returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task.

134. As detailed below in conjunction with publicly available literature published in Arizona, many of the Plaintiffs' products and services embody all of the elements of Claim 1, as well as claims 1-10 and 12-14 of the '777 Patent.

135. As a result of the Plaintiffs' infringement of the '777 Patent, Swarm has incurred substantial monetary and other damages.

136. Plaintiffs are building their future on the back of Mr. Íñiguez' novel computer architecture. The widely recognized problem of controlling multiple IoT devices has been solved by Alfonso Íñiguez. The Patents-in-Suit directly address many of the challenges faced by today's software developers, and Plaintiffs know this.

137. 35 U.S.C. § 271(a) provides that whoever "makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention" infringes the patent. As described below, the Claim Charts demonstrate direct infringement of the Patents-in-Suit.

138. 35 U.S.C. § 271(b) provides that "[w]hoever actively induces infringement of a patent shall be liable as an infringer." Inducement often involves a showing that the alleged inducer knew of the patent, knowingly induced the infringing acts, and possessed a specific intent to encourage another's infringement of the patent. As described below, Plaintiffs were either aware of or willfully blind to the Patents-in-Suit, for example, as a result of pre-suit correspondence between Swarm and Plaintiffs.

139. 35 U.S.C. § 271(b) provides that whoever "offers to sell or sells within the United States or imports into the United States a component of a patented machine, manufacture, combination or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the invention, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a

staple article or commodity of commerce suitable for substantial noninfringing use, shall be liable as a contributory infringer."

140. Upon information and belief, early discovery will reveal facts and circumstances confirming that Plaintiffs and others made, used, sold, or offered for sale at least a material part of Swarm's inventions knowing that they would be used in the Infringing Products. Moreover, Plaintiffs' detailed product literature evidences a specific intent to encourage others to participate in the infringement of Swarm's Patents.

THE '004 PATENT

I. SWARM INVENTED NEW AND USEFUL PROCESSES AND MACHINES WHICH IMPROVE THE OPERATION OF A COMPUTER AND WHICH INCLUDE INVENTIVE CONCEPTS.

141. Claim 1 of the '004 Patent is set forth below in its entirety:

> A processing system, comprising:
>
> > a task pool;
> >
> > a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;
> >
> > a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller; and
> >
> > a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second co-processor; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller;
> >
> > wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the

> processing system on a plug-and-play basis without any communication with the controller.

A. <u>Swarm Invented a New Processing Architecture.</u>

142. The preamble of Claim 1 recites:

> A processing system, comprising: ('004; 14:10).

143. The '004 specification describes various processing systems, for example in the context of:

> [A] a distributed processing system 10 includes a single or multi-core CPU 11 and one or more solidarity or co-processing cells 12A-12 configured to communicate with a task pool 13 ... ('004; 4:31-34); and

> [A] parallel processing computer architecture [including] a CPU configured to populate a task pool, and one or more co-processors configured to proactively retrieve threads (tasks) from the task pool. ('004; 2:11-14).

144. The claimed processing system involves new and useful machines and processes, and new and useful improvements to machines and processes. Taken together, the controller, task pool, and co-processors confer a substantial advantage over prior art processing systems by allowing different types of co-processors to interact with the task pool without significantly compromising their individual performance. (*See Visual Memory LLC v. NVIDIA Corporation*, 867 F.3d 1253, 1256-1257 (Fed. Cir. 2017)). Claim 1 thus focuses on improvements to computer functionality, as opposed to merely being directed to an abstract idea. (*See Enfish, LLC v. Microsoft Corp.,* 822 F.3d 1327, 1336 (Fed. Cir. 2016)). Moreover, these improvements are agnostic to the *type* of activities (whether human or non-human) to be processed.

145. Even assuming, *arguendo*, that Claim 1 may be directed to an abstract idea involving organizing human activities such as a scrum board (it is not), Claim 1 nonetheless includes inventive concepts that amount to significantly more than an abstract idea. For example, each co-processor may be configured to retrieve a task by sending its agent to the

task pool when the co-processor is idle or otherwise able to contribute processing cycles without impeding its normal operation. In this context, the term agent refers to a software module, analogous to a network packet, associated with a co-processor that interacts with the task pool to obtain tasks which are appropriate for that co-processor. ('004; 3:1-14). Humans are not capable of performing tasks such as transmitting a network packet from a co-processor to a data structure (*e.g.*, task pool); they are specific to computer operation.

B. Swarm Invented a New Processing Architecture Comprising a Task Pool Interposed Between the CPU and the Co-Processors.

146. Claim 1 further recites:

> a task pool ('004; 14:11).

147. The '004 specification describes the new processing architecture in terms of the interaction among the task pool, the controller (CPU), and the co-processors:

> The co-processors may also be capable of acting autonomously; that is, they may interact with the task pool independently of the CPU. In a preferred embodiment, each co-processor includes an agent that interrogates the task pool to seek a task to perform. As a result, the co-processors work together "in solidarity" with one another and with the task pool to complete aggregate computational requirements by autonomously retrieving and completing individual tasks which may or may not be inter-related. ('004; 2:20- 28).

148. The task pool improves the operation of a computer by acting as an intermediary device between the CPU and the co-processors. More particularly, conventional processors include a CPU and one or more co-processors, where "[t]he CPU partitions the computational requirements into tasks and distributes the tasks to co-processors." ('004; 1:56-59). Consequently, "a significant amount of CPU bandwidth is consumed by task distribution; waiting for tasks to be completed before distributing new tasks (often with dependencies on previous tasks); responding to interrupts from co-processors when a task is completed; and responding to other messages from co-processors." ('004 1:63-2:1).

149. To address these shortcomings, Swarm invented a revolutionary new parallel processing paradigm, including co-processors configured to proactively retrieve new tasks from the task pool without having to communicate directly with (or wait for) the CPU.

150. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. (*See Aatrix Software, Inc. v. Green Shades Software, Inc.*, 882 F.3d 1121, 1128 (Fed. Cir. 2018)). For example, the CPU may be programmed "to recognize and communicate with the task pool 13 and divide the computing requirements into threads ...." ('004; 5:40-43). As a result, "a co-processor may interact with the task pool without being instructed to do so by the CPU or by the task pool" ('004; 2:38-40).

C. **<u>Swarm Invented a New Processing Architecture Including a Controller Configured to Place Tasks Into the Task Pool.</u>**

151. Claim 1 further recites:

> a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks ('004; 14:12-13).

152. The '004 specification describes various controllers (CPUs), for example in the context of the multi-processor networks illustrated in FIGS. 1 and 4:

> Referring now to FIG. 4, an internet of things network 400 includes a controller (CPU) 402, a task pool 408, and various devices 410-422, some or all of which include an associated or embedded microcontroller, such as an integrated circuit (IC) chip or other component which embodies processing capacity. ('004; 11:35-40);
>
> ...
>
> In the illustrated embodiment, the controller 402 may be a smartphone, tablet, laptop, or other device which may include a display 404 and a user interface (*e.g.*, keypad) 406 for facilitating user interaction with the various devices on the network. ('004; 11:46-50);
>
> ...

> For example, in FIG. 1, the system 10 may divides an aggregate computational problem into a group of tasks, and populate the task pool 13 with a first type, a second type, and a third type of tasks. ('004, 6:39-42)

153. Claim 1 is directed to improvements to computer functionality because the controller's operating code is specifically programmed to cause the controller to distribute tasks to a task pool, as opposed to conventional processing systems in which the controller distributes tasks directly to the co-processors.

154. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. For example, "the CPU 11 may be configured for use within the system 10 by programming it to recognize and communicate with the task pool 13 and divide the computing requirements into threads." ('004, 5:40-43). By using the task pool as an intermediary device between the controller and the co-processors, the elements of Claim 1, both individually and as a combination, specifically prevent and override the routine and conventional sequence of events performed by prior art processing architectures. (*See SRI Int'l, Inc. v. Cisco Sys., Inc.*, 918 F.3d 1368 (Fed. Cir. 2019) (quoting *DDR Holdings, LLC v. Hotels.com, L.P.,* 773 F.3d 1245, 1257 (Fed. Cir. 2014))).

D. Swarm Invented a New Processing Architecture Comprising First and Second Co-Processors, Each Configured to Retrieve Tasks From the Task Pool Rather Than From the CPU.

155. Claim 1 further recites:

> a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller ('004; 14:14-19); and
>
> a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second co-processor; process the second task; generate second

> resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller ('004; 14:21-27).

156. The '004 specification describes the configuration and operation of the first and second co-processors:

> Various embodiments of a parallel processing computing architecture include a CPU configured to populate a task pool, and one or more co-processors configured to proactively retrieve threads (tasks) from the task pool. Each co-processor notifies the task pool upon completion of a task, and pings the task pool until another task becomes available for processing. In this way, the CPU communicates directly with the task pool, and communicates indirectly with the co-processors through the task pool. ('004; 2:20- 28);
>
> ...
>
> Upon retrieving a task from the task pool, a cell may then process that task, typically by retrieving data from a particular location in first memory 304, processing that data, and storing the processed data at a particular location within second memory 306. When a task is completed, the cell notifies the task pool, the task pool marks the task as completed, and the task pool notifies the CPU that the task is completed. ('004; 11:21-28);
>
> ...
>
> Significantly, the retrieval of tasks and the processing of data by the cells may occur without direct communication between the CPU and the various cells. ('004; 11:31-34).

157. The first and second co-processors, both individually and in combination with each other and/or one or more additional co-processors, improve the operation of a computer by retrieving tasks from a task pool (rather than from the CPU). The co-processors further improve the operation of computers by updating the task pool to reflect task completion, as opposed to conventional processing architectures in which the co-processors directly update the CPU.

158. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the first

and second co-processors are specifically programmed to retrieve respective tasks from the task pool, and subsequently update the task pool after completing their respective tasks, without directly communicating with the controller.

159. Moreover, the specification refers to the co-processors as autonomous, proactive solidarity cells. In this context, the term autonomous implies that a co-processor may interact with the task pool without being instructed to do so by the CPU or by the task pool. The term proactive suggests that each co-processor may be configured (*e.g.*, programmed) to periodically send an agent to monitor the task pool for available tasks appropriate to that co-processor. The term solidarity implies that co-processing cells share a common objective in monitoring and executing all available tasks within the task pool. Prior to Swarm's invention, these inventive concepts had never been proposed before, and thus they involve more than well-understood, routine, and conventional activities previously known to the industry.

E. <u>Swarm Invented a New Processing Architecture Configured to Dynamically Accept the First, Second, and an Additional Co-Processor on a Plug-and-Play Basis.</u>

160. Claim 1 further recites:

> wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller. ('004; 14:28-32).

161. The '004 specification describes the dynamic plug-and-play feature of the invention:

> [I]nteroperability among the CPU and co-processors may be facilitated by configuring the CPU to compose and/or structure tasks at a level of abstraction which is independent of the instruction set architecture associated with the various co-processors, thereby allowing the components to communicate at a task level rather than at an instruction level. As such, devices and their associated co-processors may be added to a network on a 'plug and play' basis. ('004; 3:34-40).

162. Dynamically accepting co-processors on a plug-and-play basis improves the operation of a computer network by integrating co-processors with different instruction set architectures into the same network. ('004; 3:43-44).

163. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the system may include a plurality of cells, wherein some of the cells are capable of performing the same task types as other cells, to thereby create redundancy in the system. This redundancy allows the system to continue functioning seamlessly when cells are removed from the system or are otherwise unavailable. The system also functions seamlessly when cells are dynamically added to the system. ('004; 6:34-55). These inventive concepts had never been proposed before Swarm invented them.

164. Accordingly, Claim 1 is directed to a new processing architecture which improves the operation of computer, and which includes significantly more than well-understood, routine, and conventional activities.

165. Claims 2 – 12 of the '004 Patent are also directed to various features of a new processing architecture which improve the operation of computer, and which include significantly more than well-understood, routine, and conventional activities.

166. By way of non-limiting example, Claim 4 is directed to a processing system "wherein the controller and the task pool reside on a *monolithic integrated circuit* (IC), and the first and second co-processors do not reside on the IC."

167. Claim 6 is directed to a processing system "wherein the first and second devices each comprise one of a sensor, light bulb, power switch, appliance, biometric device, medical device, diagnostic device, lap top, tablet, smartphone, motor controller, and security device."

168. Claim 7 is directed to a processing system "wherein the first co-processor is configured to modify a task within the task pool."

169. Claim 8 is directed to a processing system wherein "the first co-processor is further configured to process the first and notify the task pool upon completion of the first task."

170. Claim 9 is directed to a processing system wherein "the task pool is configured to notify the controller upon completion of the first task."

171. Claim 10 is directed to a processing system "wherein the controller is configured to communicate with the first co-processor and the second co-processor only indirectly through the task pool."

172. Claim 11 is directed to a processing system "wherein the first co-processor is configured to deposit a new task into the task pool."

173. Claim 12 is directed to a processing system "wherein the first co-processor is configured to determine when it has available processing capacity, and to dispatch the first agent to the task pool in response to the determination."

174. As explained in detail in the '004 specification, each of the foregoing claims are directed to improvements to the operation of computers, and include significantly more than well-understood, routine, and conventional activities.

THE '275 PATENT

II. SWARM INVENTED NEW AND USEFUL PROCESSES AND MACHINES WHICH IMPROVE THE OPERATION OF A COMPUTER AND WHICH INCLUDE INVENTIVE CONCEPTS.

175. Claim 1 of the '275 Patent is set forth below in its entirety:

> A collaborative intelligence system, comprising:
>
> a task pool;
>
> a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;
>
> a first co-processor configured to successively: proactively retrieve a first task from the task pool; process the first task; generate first resulting data; and update the task pool to reflect completion of the first

> task, all without any communication between the first co-processor and the controller; and
>
> a second co-processor configured to successively: proactively retrieve a second task from the task pool; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller;
>
> wherein the collaborative intelligence system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller;
>
> the plurality of first tasks and the plurality of second tasks are associated with a common objective;
>
> the first and second co-processors autonomously work together in solidarity with the task pool to complete the common objective.

A. <u>Swarm Invented a New Processing Architecture.</u>

176. The preamble of Claim 1 recites:

> A collaborative intelligence system, comprising: ('275; 14:24).

177. The '275 specification describes various collaborative intelligence systems, for example in the context of:

> [P]arallel processing computing systems and environments (such as IoT and collaborative intelligence environments), ranging from simple switching and control functions to complex programs and algorithms including, without limitation: robot control, data encryption; graphics, video, and audio processing; direct memory access; mathematical computations; data mining; game algorithms; ethernet packet and other network protocol processing including construction, reception and transmission of data the outside network; financial services and business methods; search engines; internet data streaming and other web-based applications; execution of internal or external software programs; switching on and off and/or otherwise controlling or manipulating appliances, light bulbs, consumer electronics, robotic vehicles,

> and the like, *e.g.*, in the context of the Internet-of-Things and/or collaborative intelligence systems. ('275; 4:18-34).

178. The claimed collaborative intelligence system involves new and useful machines and processes, and new and useful improvements to machines and processes. Taken together, the controller, task pool, and co-processors confer a substantial advantage over prior art processing systems by allowing different types of co-processors to interact with the task pool without significantly compromising their individual performance. Claim 1 is thus directed to improvements to computer functionality, as opposed to merely being directed to an abstract idea. Moreover, these improvements are agnostic to the *type* of activities (whether human or non-human) to be processed.

179. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea involving organizing human activities such as a scrum board (it is not), Claim 1 nonetheless includes inventive concepts that amount to significantly more than an abstract idea. For example, each co-processor may be configured to retrieve a task by sending its agent to the task pool when the co-processor is idle or otherwise able to contribute processing cycles without impeding its normal operation. In this context, the term agent refers to a software module, analogous to a network packet, associated with a co-processor that interacts with the task pool to obtain tasks which are appropriate for that co-processor. ('275; 3:21-24). Humans are not capable of performing tasks such as transmitting a network packet from a co-processor to a data structure (*e.g.*, task pool), as they are specific to computer operation.

B. <u>Swarm Invented a New Processing Architecture Comprising a Task Pool Interposed Between the CPU and the Co-Processors.</u>

180. Claim 1 further recites:

> a task pool ('275; 14:25).

181. The '275 specification describes the new processing architecture in terms of the interaction among the task pool, the controller (CPU), and the co-processors:

> The co-processors may also be capable of acting autonomously; that is, they may interact with the task pool independently of the CPU. In a preferred embodiment, each co-

> processor includes an agent that interrogates the task pool to seek a task to perform. As a result, the co-processors work together "in solidarity" with one another and with the task pool to complete aggregate computational requirements by autonomously retrieving and completing individual tasks which may or may not be inter-related. ('275; 2:28-36).

182. The task pool improves the operation of a computer by electronically communicating with the CPU as well as the co-processors. More particularly, conventional processors include a CPU and one or more co-processors, where "[t]he CPU partitions the computational requirements into tasks and distributes the tasks to co-processors." ('275; 1:63-64). Consequently, "a significant amount of CPU bandwidth is consumed by task distribution; waiting for tasks to be completed before distributing new tasks (often with dependencies on previous tasks); responding to interrupts from co-processors when a task is completed; and responding to other messages from co-processors." ('275 2:3-8).

183. To address these shortcomings, Swarm invented a revolutionary new parallel processing paradigm, including co-processors configured to proactively retrieve new tasks from the task pool without having to communicate directly with (or wait for) the CPU.

184. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. For example, the CPU may be programmed "to recognize and communicate with the task pool 13 and divide the computing requirements into threads…." ('275; 5:54-56). As a result, "a co-processor may interact with the task pool without being instructed to do so by the CPU or by the task pool" ('275; 2:46-48).

C. <u>**Swarm Invented a New Processing Architecture Comprising a Controller Configured to Place Tasks Into the Task Pool.**</u>

185. Claim 1 further recites:

> a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks ('275; 14:26-27).

186. The ’275 specification describes various controllers (CPUs), for example in the context of the multi-processor networks illustrated in FIGS. 1 and 4:

> Referring now to FIG. 4, an internet of things network 400 includes a controller (CPU) 402, a task pool 408, and various devices 410-422, some or all of which include an associated or embedded microcontroller, such as an integrated circuit (IC) chip or other component which embodies processing capacity. (’275; 11:51-56);
>
> ...
>
> In the illustrated embodiment, the controller 402 may be a smartphone, tablet, laptop, or other device which may include a display 404 and a user interface (*e.g.*, keypad) 406 for facilitating user interaction with the various devices on the network. (’275; 11:62-66);
>
> ...
>
> For example, in FIG. 1, the system 10 may divides an aggregate computational problem into a group of tasks, and populate the task pool 13 with a first type, a second type, and a third type of tasks. (’275, 6:54-57).

187. Claim 1 is directed to improvements to computer functionality because the controller’s operating code is specifically programmed to cause the controller to distribute tasks to the task pool, as opposed to conventional processing systems in which the controller distributes tasks directly to the co-processors.

188. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. For example, “the CPU 11 may be configured for use within the system 10 by programming it to recognize and communicate with the task pool 13 and divide the computing requirements into threads.” (’275, 5:54-56). By using the task pool as an intermediary device between the controller and the co-processors, the elements of Claim 1, both individually and as a combination, specifically prevent and override the routine and conventional sequence of events performed by prior art processing architectures.

D. **Swarm Invented a New Processing Architecture Comprising First and Second Co-Processors, Each Configured to Coordinate Tasks with the Task Pool instead of the CPU.**

189. Claim 1 further recites:

> a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller ('275; 14:28-33); and
>
> a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second co-processor; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller. ('275; 14:34-39).

190. The '275 specification describes the configuration and operation of the first and second co-processors:

> Various embodiments of a parallel processing computing architecture include a CPU configured to populate a task pool, and one or more co-processors configured to proactively retrieve threads (tasks) from the task pool. Each co-processor notifies the task pool upon completion of a task, and pings the task pool until another task becomes available for processing. In this way, the CPU communicates directly with the task pool, and communicates indirectly with the co-processors through the task pool. ('275; 2:19-27);
>
> ...
>
> Upon retrieving a task from the task pool, a cell may then process that task, typically by retrieving data from a particular location in first memory 304, processing that data, and storing the processed data at a particular location within second memory 306. When a task is completed, the cell notifies the task pool, the task pool marks the task as completed, and the task pool notifies the CPU that the task is completed. ('275; 11:37-44);
>
> ...

> Significantly, the retrieval of tasks and the processing of data by the cells may occur without direct communication between the CPU and the various cells. ('275; 11:47-50).

191. The first and second co-processors, both individually and in combination with each other and/or one or more additional co-processors, improve the operation of a computer by retrieving tasks from a task pool (rather than from the CPU). The co-processors further improve the operation of computers by updating the task pool to reflect task completion, as opposed to conventional processing architectures in which the co-processors directly update the CPU.

192. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the first and second co-processors are specifically programmed to retrieve respective tasks from the task pool, and subsequently update the task pool after completing their respective tasks, without directly communicating with the controller.

193. Moreover, the specification refers to the co-processors as autonomous, proactive solidarity cells. In this context, the term autonomous implies that a co-processor may interact with the task pool without being instructed to do so by the CPU or by the task pool. The term proactive suggests that each co-processor may be configured (*e.g.*, programmed) to periodically send an agent to monitor the task pool for available tasks appropriate to that co-processor. The term solidarity implies that co-processing cells share a common objective in monitoring and executing all available tasks within the task pool. Prior to Swarm's invention, these inventive concepts had never been proposed before, and thus they involve more than well-understood, routine, and conventional activities previously known to the industry.

E. <u>**Swarm Invented a New Processing Architecture Configured to Dynamically Accept the First, Second, and an Additional Co-Processor on a Plug-and-Play Basis.**</u>

194. Claim 1 further recites:

> herein the collaborative intelligence system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller ('275; 14:40-44).

195. The '275 specification describes the dynamic plug-and-play feature of the invention:

> [I]nteroperability among the CPU and co-processors may be facilitated by configuring the CPU to compose and/or structure tasks at a level of abstraction which is independent of the instruction set architecture associated with the various co-processors, thereby allowing the components to communicate at a task level rather than at an instruction level. As such, devices and their associated co-processors may be added to a network on a 'plug and play' basis. ('275; 3:42-50).

196. Dynamically accepting co-processors on a plug-and-play basis improves the operation of a computer network by integrating co-processors with different instruction set architectures into the same network. ('275; 3:42-52).

197. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the system may include a plurality of cells, wherein some of the cells are capable of performing the same task types as other cells, to thereby create redundancy in the system. This redundancy allows the system to continue functioning seamlessly when cells are removed from the system or are otherwise unavailable. The system also functions seamlessly when cells are dynamically added to the system. ('275; 6:49-7:2) These inventive concepts had never been proposed before Swarm invented them.

F. <u>Swarm Invented a New Processing Architecture in Which the First and Second Tasks are Associated with a Common Objective.</u>

198. Claim 1 further recites:

> the plurality of first tasks and the plurality of second tasks are associated with a common objective ('275; 14:45-46).

199. The ’275 specification describes the relationship of the first and second tasks to a common objective:

> The term solidarity implies that co-processing cells share a common objective in monitoring and executing all available tasks within the task pool. (’275; 2:51-54).

200. Associating the first and second tasks with a common objective improves the operation of a computer network by promoting swarm (or collaborative) intelligence. (’275; 1:1).

201. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the invention facilitates collaborative intelligence through the use of dynamically configurable proactive autonomous agents. (’275; 1:2-4).

G. <u>Swarm Invented a New Processing Architecture Comprising First and Second Co-Processors Which Autonomously Work Together in Solidarity with the Task Pool to Complete the Common Objective.</u>

202. Claim 1 further recites:

> the first and second co-processors autonomously work together in solidarity with the task pool to complete the common objective (’275; 14:47-49).

203. The ’275 specification describes the autonomous action of the co-processors:

> The present invention generally relates to parallel-process computing, and collaborative intelligence, and particularly relates to a processing architecture which involves autonomous co-processors (such as robotic vehicles, Internet of Things (IoT) components, and networked devices) configured to proactively retrieve tasks from a task pool populated by a central processing unit. (’275; 1:17-23).

204. By autonomously working together in solidarity with the task pool to complete the common objective, the first and second co-processors improve the operation of a computer network by effectively harnessing and exploiting available co-processing resources. (’275; 2:14-15).

205. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, by more effectively harnessing available co-processing resources, the invention reduces CPU management overhead. ('275; 2:13). These inventive concepts had never been proposed before Swarm invented them.

206. Accordingly, Claim 1 of the '275 Patent is directed to a new processing architecture which improves the operation of computer, and which includes significantly more than well-understood, routine, and conventional activities.

207. Claims 2 – 17 of the '275 Patent are also directed to various features of a new processing architecture which improve the operation of computer, and which include significantly more than well-understood, routine, and conventional activities.

208. By way of non-limiting example, Claim 2 is directed to a collaborative processing system "wherein the first co-processor is configured to perform at least one of depositing a new task into the task pool and modifying an existing task within the task pool."

209. Claim 4 is directed to a collaborative processing system "wherein the first co-processor is configured to determine when it has available processing capacity, and to dispatch the first agent to the task pool in response to the determination."

210. Claim 5 is directed to a collaborative processing system "wherein the controller and the task pool reside on a monolithic integrated circuit (IC) that is not a component of either the first or second co-processors."

211. Claim 6 is directed to a collaborative processing system wherein:

> when the first agent is retrieving the first task from the task pool, the first source address corresponds to an address associated with the first co-processor, the first destination address corresponds to an address associated with the task pool, and the first payload includes a first function which the first co-processor is configured to perform;
>
> when the first agent is returning from the task pool, the first source address is the task pool's address, the first destination

> address is the first co-processor's address, and the first payload includes a descriptor of the first task;

212. Claim 8 is directed to a collaborative processing system "wherein the first co-processor is an unmanned autonomous vehicle configured to operate as at least one of a ground vehicle and an aerial vehicle in connection with defense field operations."

213. Claim 10 is directed to a collaborative processing system "wherein neither the controller nor the task pool are incorporated into either the first or second co-processor."

214. Claim 16 is directed to a collaborative processing system "wherein the plurality of autonomous co-processors retrieve the tasks from the task pool via a wireless data connection."

215. Claim 17 is directed to a collaborative processing system wherein "at least one of the plurality of co-processors includes an agent configured to instruct the task pool to select a task of a compatible type and deliver the selected task to the at least co-processors."

216. As explained in detail in the '275 specification, each of the foregoing claims are directed to improvements to the operation of computer, and include significantly more than well-understood, routine, and conventional activities.

THE '777 PATENT

III. SWARM INVENTED NEW AND USEFUL PROCESSES AND MACHINES WHICH IMPROVE THE OPERATION OF A COMPUTER AND WHICH INCLUDE INVENTIVE CONCEPTS.

217. Claim 1 of the '777 Patent is set forth below in its entirety:

> An apparatus for parallel processing of a large computing requirement, the apparatus comprising:
>
> a central processing unit ("CPU");
>
> a task pool in electronic communication with the CPU; and

> a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process;
>
> wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks;
>
> wherein each task comprises a descriptor, the descriptor containing at least:
>
> a function to be executed; and
>
> a memory location of data upon which the function is to be executed;
>
> wherein the first agent is a data frame comprising:
>
> a source address, a destination address and a payload;
>
> wherein the first agent retrieves the matching task by:
>
> being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform;
>
> searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and
>
> returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task.

A. <u>**Swarm Invented a New Processing Architecture.**</u>

218. The preamble of Claim 1 recites:

> An apparatus for parallel processing of a large computing requirement, the apparatus comprising: ('777; 7:41-42).

219. The '777 specification describes various parallel processing embodiments, for example in the context of co-processors which work in solidarity:

> to complete a large computational requirement by processing threads and subtasks. ('777; 1:53-55).

220. The claimed apparatus involves new and useful machines and processes, and new and useful improvements to machines and processes. Taken together, the CPU, task pool, and solidarity cell provide substantial advantage over prior art processing systems by dividing a large computational requirement into a plurality of threads and placing the threads in the task pool. Claim 1 is thus directed to improvements to computer functionality, as opposed to merely being directed to an abstract idea. Moreover, these improvements are agnostic to the *type* of activities (whether human or non-human) to be processed.

221. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea involving organizing human activities such as a scrum board (it is not), Claim 1 nonetheless includes inventive concepts that amount to significantly more than an abstract idea. For example, Claim 1 comprises a task pool in electronic communication with the CPU, and a first solidarity cell in electronic communication with the task pool. Humans are not capable of electronically communicating with microelectronic devices; rather, such electronic communication is the exclusive domain of computers and computer networks.

B. <u>Swarm Invented a New Processing Architecture Comprising a Task Pool Interposed Between the CPU and the Co-Processors.</u>

222. Claim 1 further recites:

> a central processing unit ("CPU"); ('777; 7:41-43).

223. The '777 specification describes the new processing architecture in terms of the interaction among the task pool, the controller (CPU), and the co-processors:

> A method and apparatus for processing information in parallel uses autonomous computer processing units, referred to herein as solidarity cells, to process instructions intended to be executed by a central processing unit ("CPU"). ('777; 1:59-62).
>
> ...

> The CPU divides the information into one or more tasks. A task may include task threads, which each contain one or more subtasks to be performed. The CPU transmits the tasks to a task pool. Each solidarity cell in the system is connected, physically or wirelessly, to the task pool through a switching fabric. The switching fabric facilitates connections for data transfer and arbitration between all system resources. Each solidarity cell is proactive, in that it obtains a task to perform by sending its agent to the task pool when the solidarity cell has no processing to perform. The agent is a software module that searches the task pool for available tasks that match the cell's instruction set architecture. The solidarity cells may execute the task threads sequentially or in parallel, and independently or collaboratively, depending on recipes provided by the CPU. Interdependent tasks within the task pool may be logically combined as needed by the recipe. The task pool notifies the CPU when a task thread is completed. ('777; 2:1-18).

224. The central processing unit improves the operation of a computer by placing the tasks in the task pool, as opposed to sending them directly to the solidarity cells. More particularly, one drawback of conventional multiprocessing frameworks surrounds the architectural requirement that the CPU divide and distribute the threads to the co-processors. Consequently, a significant amount of the CPU's processing time is consumed in managing the co-processing tasks including: distributing the tasks, in sequential order when needed, to co-processors according to their capabilities; waiting for tasks to be completed before distributing result-dependent threads; responding to interrupts from co-processors every time a task is completed; and responding to other messages from co-processors." ('777; 1:31-41).

225. To address these shortcomings, Mr. Íñiguez invented a revolutionary new parallel processing paradigm, including co-processors configured to proactively retrieve new tasks from the task pool without having to communicate directly with (or wait for) the CPU. This alleviates the management workload on the CPU, while keeping the co-processors busy. ('777; 1:43-45).

226. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. For example, the CPU may be programmed "to recognize and communicate with the task pool 13 and divide the computing requirements into threads … ." ('777; 3:17-19).

C. <u>Swarm Invented a New Processing Architecture Comprising a Task Pool in Electronic Communication with the CPU.</u>

227. Claim 1 further recites:

> a task pool in electronic communication with the CPU ('777; 7:44).

228. The '777 specification describes a task pool in electronic communication with the CPU, for example in the context of:

> The CPU 11 may communicate with the task pool 13 directly or through the switching fabric 14. ('777; 2:56-57);
>
> ...
>
> The ability of the CPU 11 to perform the inventive parallel processing methods within the presently described system 10 depends on the CPU's 11 operating system. Specifically, the CPU 11 is a suitable CPU for the system 10 if its operating system may be programmed to recognize and communicate with the task pool 13 and divide computing requirements into threads as described below. ('777; 3:13-19).

229. Claim 1 is directed to improvements to computer functionality in part because the task pool is in electronic communication with the CPU, as opposed to conventional processing systems in which the CPU distributes tasks directly to the co-processors.

230. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes inventive concepts involving more than well-understood, routine, and conventional activities previously known to the industry. For example, the CPU's operating system may be specifically programmed to recognize and communicate with the task pool. ('777; 3:13-19). By using the task pool as an intermediary

device between the CPU and the solidarity cell, the claimed invention specifically prevents and overrides the routine and conventional sequence of events performed by prior art processing architectures.

D. **Swarm Invented a New Processing Architecture Comprising a Solidarity Cell Configured to Retrieve a Matching Task From the Task Pool Without Requiring an Instruction from the CPU.**

231. Claim 1 further recites:

> a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process; ('777; 7:45-49).

232. The '777 specification describes the configuration and operation of the first solidarity cell, for example in the context of:

> an apparatus and method for parallel processing in a multiprocessor system using co-processors that proactively seek threads to process. It is a further object that the co-processors be capable of acting autonomously. It is a further object that the co-processors include an agent that searches a task pool to acquire tasks for the co-processors to perform. ('777; 1:47-52).
>
> ...
>
> In particular, the cells 12A ... n do not require an instruction from the CPU 11 to act ('777; 5:21-22).

233. The solidarity cell improves the operation of a computer by proactively retrieving a matching task from the task pool. The solidarity cell further improves the operation of computers by retrieving a matching task from the task pool without requiring an instruction from the CPU.

234. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the solidarity cell is equipped with a software agent to retrieve a matching task from the task pool, without directly communicating with the CPU.

235. Prior to Swarm's invention, this inventive concept had never been proposed before, and thus Claim 1 involves more than well-understood, routine, and conventional steps.

E. <u>Swarm Invented a New Processing Architecture in Which the CPU Populates the Task Pool with a Matching Task.</u>

236. Claim 1 further recites:

> wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks; (''777; 7:50-53).

237. The '777 specification describes the matching task:

> In another embodiment, the agent 30A searches the task 22 descriptors for an executable instruction that matches one of the instructions that that cell 12A is capable of executing. When a matching task 22 is found, the agent 30A delivers the descriptor of the matching task 22 to the cell 12A, which begins to process the task 22 ('777; 6:25-31).

238. The CPU improves the operation of a computer network by dividing the requirement into one or more threads, and placing the tasks (including the matching task) in the task pool.

239. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the agent searches the task pool for an executable instruction that matches one of the instructions that that particular solidarity cell is capable of executing. When a matching task 22 is found, the agent delivers the descriptor of the matching task to its solidarity cell. This inventive concept had never been proposed before Swarm's invention.

F. <u>Swarm Invented a New Processing Architecture Comprising a Task Format Which Includes a Descriptor Defining a Function to be Executed and a Location of the Data Upon Which the Function is to be Executed.</u>

240. Claim 1 further recites:

> wherein each task comprises a descriptor, the descriptor containing at least:
>
> a function to be executed; and
>
> a memory location of data upon which the function is to be executed; ('777; 7:54-58).

241. The '777 specification describes the sub-parts of an exemplary task:

> The descriptor may contain one or more of a specific instruction to be executed, a mode of execution, the location of the data to be processed, and the location for placement of the results, if any. ('777; 4:41-44)
>
> ...
>
> In the preferred embodiment, the descriptor is a data structure containing a header and a plurality of reference pointers to memory locations, and the task 22 includes the memory address of the data structure. The header defines the function or instruction to execute. A first pointer references the location of the data to be processed. ('777; 4:50-56).

242. Defining a common task format improves the operation of a computer network by enabling different types of specialized solidarity cells, which may not be directly compatible with each other, to efficiently process large computational requirements within a heterogeneous computing environment. ('777; 3:49-56).

243. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, Claim 1 states that a task may include a descriptor comprising: i) a function to be executed; and ii) a memory location of data upon which the function is to be executed ('777; 7:54-58). This inventive concept had never been proposed before Swarm's invention.

G. <u>Swarm Invented a New Processing Architecture Including a Data Frame Comprising a Source Address, a Destination Address, and a Payload.</u>

244. Claim 1 further recites:

> wherein the first agent is a data frame comprising:
>
> a source address, a destination address and a payload; ('777; 7:59-60).

245. The ’777 specification describes an agent in a data frame format:

> An agent 30A, B, C, D ... n, hereinafter collectively referred to as agent 30A ... n to indicate that the system 10 has the same number of agents as solidarity cells 12A ... n, may be considered a data frame in the networking sense. It contains a source address, a destination address, and a payload. (’777; 5:28-33).

246. The claimed agent improves the operation of a computer by selectively retrieving only those tasks which are appropriate for that particular agent.

247. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, a large computing requirement may be parsed into tasks of “a first type, a second type, and a third type; a first cell 12A is capable of performing only tasks of the first type; a second cell 12B can perform tasks of the second type; a third cell 12C can perform tasks of the third type; a fourth cell 12D can perform tasks of the second or third types; and a fifth cell 12N can perform all three task types.” (’777; 3:62-4:2). This inventive concept had never been proposed before Swarm’s invention.

H. Swarm Invented a New Processing Architecture Comprising a Solidarity Cell Configured to Retrieve a Matching Task by Dispatching an Agent to the Task Pool.

248. Claim 1 further recites:

> wherein the first agent retrieves the matching task by:
>
> > being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform; (’777; 7:61-67).

249. The ’777 specification describes an agent being dispatched by its solidarity cell to the task pool:

> To acquire a task 22, a cell 12A sends an agent 30A to the task pool 13 to search for and retrieve an available task 22 that

> requires completion, is not locked, and has a task type that can be performed by the cell 12A. ('777; 5:25-28).
>
> ...
>
> When the agent 30A ... n is dispatched to the task pool 13, the payload contains identifying information of the types of tasks the corresponding cell 12A ... n can perform. When the agent 30A ... n returns from the task pool 13, the payload contains the descriptor of the task 22, either in the form of a memory location or the full descriptor data structure. ('777; 5:55-60).

250. The solidarity cell improves the operation of a computer by dispatching an agent to the task pool, along with a payload which permits the agent to selectively retrieve tasks which are appropriate for that agent.

251. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, Claim 1 recites that each solidarity cell's payload includes "a list of functions which that solidarity cell is configured to perform." This inventive concept had never been proposed before Swarm's invention.

I. **Swarm Invented a New Processing Architecture Comprising a Solidarity Cell Configured to Retrieve a Matching Task by Selecting an Appropriate Task From the Task Pool.**

252. Claim 1 further recites:

> searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and ('777; 8:1-3).

253. The '777 specification describes an agent searching the task pool to retrieve an appropriate task:

> The agent is a software module that searches the task pool for available tasks that match the cell's instruction set architecture. ('777; 2:10-13).

254. The claimed agent improves the operation of a computer by searching the task pool for a task that has a function which the first solidarity cell can perform.

255. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, a large computing requirement may be parsed into tasks of "a first type, a second type, and a third type; a first cell 12A is capable of performing only tasks of the first type; a second cell 12B can perform tasks of the second type; a third cell 12C can perform tasks of the third type; a fourth cell 12D can perform tasks of the second or third types; and a fifth cell 12N can perform all three task types." ('777; 3:62-4:2). This inventive concept had never been proposed before Swarm's invention.

J. **Swarm Invented a New Processing Architecture Comprising a Solidarity Cell Configured to Return the Matching Task to the Cell.**

256. Claim 1 further recites:

> returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task. ('777; 8:4-8).

257. The '777 Patent specification describes an agent returning to the solidarity cell from the task pool:

> The source and destination addresses may serve two functions. First, the addresses guide transmission of the agent 30A ... n. The destination address is the address of the task pool 13 when the agent 30A ... n is seeking a task 22, and is the address of the corresponding cell 12A ... n when the agent 30A ... n is returning with a task 22. Correspondingly, the source address is the address of the cell 12A ... n when the agent 30A ... n is seeking a task 22, and is the address of the task pool 13 when the agent 30A ... n is returning with a task 22. ('777; 5:33-42).

258. The solidarity cell improves the operation of a computer by performing the following steps to retrieve a task: i) dispatching an agent to the task pool with a payload identifying the types of tasks the cell can perform; ii) searching the task pool for an appropriate task; and iii) returning to the solidarity cell with a payload identifying the descriptor of the matching task.

259. Even assuming, *arguendo*, that Claim 1 is directed to an abstract idea (it is not), Claim 1 nonetheless includes numerous inventive concepts. For example, the solidarity cell is configured such that the destination address is the address of the task pool when the agent is seeking a task, and the destination address is the address of its solidarity cell when the agent is returning with a task. In contrast, the source address is the address of the solidarity cell when the agent is seeking a task, and the source address is the address of the task pool when the agent returns a task to the cell. These inventive concepts had never been proposed before Swarm invented them.

260. Accordingly, Claim 1 of the '777 Patent is directed to a new processing architecture which improves the operation of computers, and which includes significantly more than well-understood, routine, and conventional activities.

261. Claims 2-14 of the '777 Patent are also directed to various features of a new processing architecture which improve the operation of computers, and which include significantly more than well-understood, routine, and conventional activities.

262. By way of non-limiting example, Claim 2 is directed to an apparatus "wherein the task pool notifies the CPU when the tasks of a thread are completed."

263. Claim 3 is directed to an apparatus "wherein the tasks each comprise a task type selected from a set of task types, and wherein the first solidarity cell is configured to perform tasks of one or more of the task types."

264. Claim 4 is directed to an apparatus "wherein the matching task is a task that is ready to be processed and has a task type that the first solidarity cell can perform."

265. Claim 8 is directed to an apparatus "wherein the descriptor further contains a memory location where processed data is to be stored."

266. Claim 9 is directed to an apparatus "wherein the descriptor is a data structure and the task contains a reference to the memory location of the descriptor."

267. Claim 10 is directed to an apparatus "wherein the task pool occupies a region of physical memory."

268. Claim 11 is directed to an apparatus "wherein the task pool is disposed in a hardware block dedicated to the task pool."

269. As explained in detail in the '777 specification, each of the foregoing claims are directed to improvements to the operation of computer, and include significantly more than well-understood, routine, and conventional activities.

PLAINTIFFS' PRODUCTS AND SERVICES

270. Plaintiffs' websites describe various networking products and services. Many of these products and services infringe one or more of the Patents-in-Suit either directly under 35 U.S.C. § 271(a), through inducement under § 271(b), and/or by way of contributory infringement under § 271(c).

271. For example, the web page located at https://www.juniper.net/content/dam/www/assets/datasheets/us/en/network-automation/apstra-solution.pdf reveals a plurality of product families, systems, and sub-systems, including references to Juniper Apstra 4.0.

272. The Claim Charts provide non-limiting illustrations which "map" the Patents-in-Suit to exemplary Infringing Products, and are incorporated herein.

EXEMPLARY CLAIM CHARTS

THE '004 PATENT

273. A first claim chart mapping the elements of Claims 1-3 and 7-12 of the '004 patent to the Infringing Products is attached hereto as Exhibit D. Additional references, including literature describing Plaintiffs' products and services, are cited in the Claim Charts and are attached as Exhibits G-Q.

274. More particularly, with regard to Claim 1 of the '004 Patent, the "processing system" preamble is illustrated, *inter alia*, in Appendix A of Exhibit D.

275. The "task pool" element may be found at, *inter alia*, Exhibit H, page 4; Exhibit I, page 8; and Exhibit G, page 24. Exhibits H and I were discovered in an internet search of publicly available information conducted in the ordinary course of business.

Despite being previously available on Apstra's website, as discovered, these publicly available documents had been designated as confidential. Although Swarm asserts no claim of confidentiality upon these documents, in an abundance of caution, Swarm has redacted Exhibits H and I in the event Plaintiffs wish to seek a protective order to maintain any alleged confidentiality.

276. The "controller" and "populate" elements may be found at, *inter alia*, Exhibit L, page 3; Exhibit M, page 6; and Exhibit K, page 17.

277. The "first tasks" and "second tasks" elements generally correspond to the Plaintiffs' term "intent" and/or "configuration" and may be found at, *inter alia*, Exhibit L, page 3; Exhibit M, page 6; Exhibit K, page 17; and Exhibit J, page 15.

278. The "first co-processor" element generally correspond to the Plaintiffs' term "device" and may be found at, *inter alia*, Exhibit I, page 6.

279. The "retrieve a first task from the task pool" element may be found at, *inter alia*, Exhibit H, page 4; Exhibit H, page 7; Appendix B of Exhibit D; Exhibit G, page 32; and Exhibit N, page 8.

280. The "deliver the first task to the first co-processor" element may be found at, *inter alia*, Exhibit I, page 4.

281. The "process the first task" element may be found at, *inter alia*, Exhibit J, page 15.

282. The "generate first resulting data" element may be found at, *inter alia*, Exhibit I, page 5; and Exhibit H, page 4.

283. The "and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller" element may be found at, *inter alia*, Exhibit H, page 4; and Exhibit O, page 3.

284. The various elements pertaining to the "second co-processor" which are common to the aforementioned "first co-processor" may be found at, *inter alia*, the same References cited in the above discussion of the "first co-processor."

285. The "wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller" element may be found at, *inter alia*, Exhibit H, page 4; and Exhibit I, page 7; Exhibit I, page 4; and Exhibit O, page 3.

286. In similar fashion, Exhibit D also maps the elements of Claims 2-3 and 6-12 of the '004 Patent to the Infringing Products.

THE '275 PATENT

287. A second claim chart mapping the elements of Claims 1-4, 6-7, and 9-17 of the '275 patent to the Infringing Products is attached hereto as Exhibit E. Additional references, including literature describing Plaintiffs' products and services, are cited in the Claim Charts and are attached as Exhibits G-Q.

288. More particularly, with regard to Claim 1 of the '275 Patent, the "collaborative intelligence system" preamble is illustrated, *inter alia*, in Appendix A of Exhibit E; Exhibit I, page 2; and Exhibit M, page 2.

289. The "task pool" element may be found at, *inter alia*, Exhibit H, page 4; Exhibit I, page 8; and Exhibit G, page 24.

290. The "controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks" element may be found at, *inter alia*, Exhibit L, page 3; Exhibit M, page 6; Exhibit K, page 17; and Exhibit J, page 15.

291. The "first co-processor" element may be found at, *inter alia*, Exhibit I, page 6.

292. The "proactively retrieve a first task from the task pool" element may be found at, *inter alia*, Exhibit H, page 4; Exhibit I, page 7; Exhibit H, page 7; Appendix B of Exhibit E; Exhibit G, page 32; and Exhibit N, page 8.

293. The "process the first task" element may be found at, *inter alia*, Exhibit J, page 15.

294. The "generate first resulting data" element may be found at, *inter alia*, Exhibit I, page 5; and Exhibit H, page 4.

295. The "and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller" element may be found at, *inter alia*, Exhibit H, page 4; and Exhibit O, page 3.

296. The various elements pertaining to the "second co-processor" which are common to the analogous elements pertaining to the aforementioned "first co-processor" may be found at, *inter alia*, the same References cited in the above discussion of the "first co-processor."

297. The "wherein the collaborative intelligence system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller" element may be found at, *inter alia*, Exhibit I, page 2; Exhibit M, page 2; Exhibit H, page 4; Exhibit I, page 7; Exhibit I, page 4; and Exhibit O, page 3.

298. The "plurality of first tasks and the plurality of second tasks are associated with a common objective" element may be found at, *inter alia*, Exhibit J, page 15; and Exhibit M, page 2.

299. The "first and second co-processors autonomously work together in solidarity with the task pool to complete the common objective" element may be found at, *inter alia*, Exhibit J, page 15; Exhibit L, page 3; Exhibit K, page 13; and Exhibit H, page 4.

THE '777 PATENT

300. In similar fashion, Exhibit F also maps the elements of Claims 2-4, 6-7, and 9-17 of the '777 Patent to the Infringing Products.

301. A third claim chart mapping the elements of Claims 1-10 and 12-14 of the '777 patent to the Infringing Products is attached hereto as Exhibit F. Additional references,

including literature describing Plaintiffs' products and services, are cited in the Claim Charts and are attached as Exhibits G-Q.

302. More particularly, with regard to Claim 1 of the '777 Patent, the "apparatus for parallel processing of a large computing requirement" preamble is illustrated, *inter alia*, in Appendix A of Exhibit F.

303. The "central processing unit ('CPU')" element may be found at, *inter alia*, Exhibit M, page 6.

304. The "task pool in electronic communication with the CPU" element may be found at, *inter alia*, Exhibit I, page 8; Exhibit K, page 17; Exhibit M, page 6; and Exhibit G, page 24.

305. The "first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process" element may be found at, *inter alia*, Exhibit H, page 5; Exhibit I, page 6; Exhibit H, page 4; Exhibit I, page 7; Exhibit H, page 7; Appendix B of Exhibit F; Exhibit G, page 32, Exhibit N, page 8; Exhibit O, page 3; Exhibit J, page 15; Exhibit L, page 3; and Exhibit I, page 3.

306. The "wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks" element may be found at, *inter alia*, Exhibit K, page 17; Exhibit L, page 3; Exhibit M, page 6; Exhibit J, page 15; and Exhibit I, page 3.

307. The "wherein each task comprises a descriptor, the descriptor containing at least a function to be executed" element may be found at, *inter alia*, Exhibit N, page 8; Exhibit J, page 15; and Exhibit I, page 8.

308. The "and a memory location of data upon which the function is to be executed" element may be found at, *inter alia*, Exhibit H, page 7; Exhibit N, page 8; Exhibit J, page 15; and Exhibit I, page 8.

309. The "wherein the first agent is a data frame comprising: a source address, a destination address and a payload" element may be found at, *inter alia*, Exhibit I, page 6; Exhibit J, page 37; Exhibit P, page 2; and Exhibit P, page 3.

310. The "wherein the first agent retrieves the matching task by:" may be found at, *inter alia*, Exhibit I, page 6; Exhibit H, page 4; Exhibit H, page 7; Appendix B of Exhibit F; Exhibit G, page 32; Exhibit N, page 8; Exhibit J, page 15; Exhibit L, page 3; and Exhibit I, page 3.

311. The "being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform" element may be found at, *inter alia*, Exhibit H, page 4; Exhibit J, page 37; Exhibit P, page 3; Exhibit I, page 8; and Exhibit J, page 15.

312. The "searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform" element may be found at, *inter alia*, Exhibit I, page 3; Exhibit J, page 15; and Exhibit I, page 8.

The "returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task" element may be found at, *inter alia*, Exhibit I, page 4; Exhibit J, page 37; Exhibit P, page 3; Appendix B of Exhibit F; Exhibit G, page 32; Exhibit N, page 8; Exhibit J, page 15; Exhibit L, page 3; and Exhibit I, page 3.

313. In similar fashion, Exhibit F also maps the elements of Claims 2-10 and 12-14 of the '777 Patent to the Infringing Products.

FIRST CLAIM FOR RELIEF

Infringement of the '004 Patent (35 U.S.C. § 271)

314. Swarm incorporates and realleges Paragraphs 73 through 313 of this Counterclaim.

315. Plaintiffs have infringed and continue to infringe Claims 1-3 and 7-12 of the ’004 Patent by making, using, selling, offering to sell, and/or importing infringing products and services into the United States.

316. Plaintiffs’ actions as described herein constitute direct, induced, and/or contributory infringement of the ’004 Patent in violation of 35 U.S.C § 271(a), (b), and/or (c).

317. Plaintiffs’ actions as described herein constitute infringement of the ’004 Patent either literally or under the doctrine of equivalents.

318. As a proximate result of Plaintiffs’ infringement of the ’004 Patent, Swarm has been damaged and Plaintiffs have unfairly profited in amounts to be proven at trial.

319. Plaintiffs’ infringement of the ’004 Patent has been and continues to be willful, entitling Swarm to recover treble damages and/or attorney fees pursuant to 35 U.S.C. § 284.

320. Plaintiffs’ knowing, intentional, and/or willful actions make this an exceptional case, entitling Swarm to an award of reasonable fees pursuant to 35 U.S.C. § 285.

321. Plaintiffs’ direct, inducement, and/or contributory infringement of the ’004 Patent has caused and will continue to cause Swarm irreparable harm unless they are enjoined by this Court.

SECOND CLAIM FOR RELIEF

Infringement of the ’275 Patent (35 U.S.C. § 271)

322. Swarm incorporates and realleges Paragraphs 73 through 321 of this Counterclaim.

323. Plaintiffs have infringed and continue to infringe Claims 1-4, 6-7, and 9-17 of the ’275 Patent by making, using, selling, offering to sell, and/or importing infringing products and services into the United States.

324. Plaintiffs’ actions as described herein constitute direct, induced, and/or contributory infringement of the ’275 Patent in violation of 35 U.S.C § 271(a), (b), and/or (c).

325. Plaintiffs’ actions as described herein constitute infringement of the ’275 Patent either literally or under the doctrine of equivalents.

326. As a proximate result of Plaintiffs’ infringement of the ’275 Patent, Swarm has been damaged and Plaintiffs have unfairly profited in amounts to be proven at trial.

327. Plaintiffs’ infringement of the ’275 Patent has been and continues to be willful, entitling Swarm to recover treble damages and/or attorney fees pursuant to 35 U.S.C. § 284.

328. Plaintiffs’ knowing, intentional, and/or willful actions make this an exceptional case, entitling Swarm to an award of reasonable fees pursuant to 35 U.S.C. § 285.

329. Plaintiffs’ direct, inducement, and/or contributory infringement of the ’275 Patent has caused and will continue to cause Swarm irreparable harm unless they are enjoined by this Court.

<u>THIRD CLAIM FOR RELEIF</u>

Infringement of the ’777 Patent (35 U.S.C. § 271)

330. Swarm incorporates and realleges Paragraphs 73 through 329 of this Counterclaim.

331. Plaintiffs have infringed and continue to infringe Claims 1-10 and 12-14 of the ’777 Patent by making, using, selling, offering to sell, and/or importing infringing products and services into the United States.

332. Plaintiffs' actions as described herein constitute direct, induced, and/or contributory infringement of the '777 Patent in violation of 35 U.S.C § 271(a), (b), and/or (c).

333. Plaintiffs' actions as described herein constitute infringement of the '777 Patent either literally or under the doctrine of equivalents.

334. As a proximate result of Plaintiffs' infringement of the '777 Patent, Swarm has been damaged and Plaintiffs have unfairly profited in amounts to be proven at trial.

335. Plaintiffs' infringement of the '777 Patent has been and continues to be willful, entitling Swarm to recover treble damages and/or attorney fees pursuant to 35 U.S.C. § 284.

336. Plaintiffs' knowing, intentional, and/or willful actions make this an exceptional case, entitling Swarm to an award of reasonable fees pursuant to 35 U.S.C. § 285.

337. Plaintiffs' direct, inducement, and/or contributory infringement of the '777 Patent has caused and will continue to cause Swarm irreparable harm unless they are enjoined by this Court.

AFFIRMATIVE DEFENSES

338. Swarm reserves the right to assert affirmative defenses as they become known through further investigation and discovery, including, without limitation, the doctrine of unclean hands.

PRAYER FOR RELIEF

WHEREFORE, DEFENDANT/COUNTERCLAIMANT prays for the following relief against Plaintiffs, jointly and severally:

A. A judgment that Plaintiffs have infringed one or more claims of each of the Patents-in-Suit;

B. An order and judgment temporarily and permanently enjoining Plaintiffs and their officers, directors, agents, servants, employees, affiliates, attorneys, and all others acting in privity or in concert with them, and their parents, subsidiaries, divisions, successors and assigns, from further acts of infringement;

C. A judgment awarding Swarm all damages adequate to compensate for Plaintiffs' infringement, but in no event less than a reasonable royalty, including all pre-judgment and post-judgment interest at the maximum rate permitted by law;

D. A judgment awarding Swarm all relief (including money damages) contemplated 35 U.S.C. § 154(d);

E. A judgment awarding Swarm all damages, including treble damages, based on any infringement found to be willful, pursuant to 35 U.S.C. § 284, together with prejudgment interest;

F. A judgment awarding Swarm its costs pursuant to 35 U.S.C. § 284;

G. A judgment finding that this case is exceptional and awarding Swarm its attorneys fees in accordance with 35 U.S.C. § 285; and

H. Any other remedy to which Swarm may be entitled to or the Court deems just and proper.

DEMAND FOR JURY TRIAL

Pursuant to Federal Rule of Civil Procedure 38(b), Swarm requests a trial by jury of all aspects properly triable by jury.

DATED this 4th day of January 2022.

Respectfully submitted,

By *__/s/Michael K. Kelly__*

BEUS GILBERT McGRODER PLLC
Leo R. Beus
Michael K. Kelly

STEYER LOWENTHAL BOODROOKAS ALVAREZ & SMITH LLP
Allan Steyer
Suneel Jain

Attorneys for Defendant Swarm Technology LLC

CERTIFICATE OF SERVICE

I hereby certify that on **January 4, 2022**, I caused the foregoing document to be served via the Court's CM/ECF system on all counsel of record per Local Rule CV-5(5).

__/s/Suneel Jain__
Suneel Jain

EXHIBIT LIST

Exhibit	Title
A	U.S. Patent No. 9,852,004
B	U.S. Patent No. 10,592,275
C	U.S. Patent No. 9,146,777
D	U.S. Patent No. 9,852,004 Claim Charts
E	U.S Patent No. 10,592,275 Claim Charts
F	U.S. Patent No. 9,146,777 Claim Charts
G	Juniper Apstra 4.0 User Guide
H	The AOS and the Distributed Systems Challenge in Data Center Automation
I	The AOS Architecture Overview
J	Intent-Based Networking for Dummies, Apstra Special Edition
K	Juniper Apstra Architecture
L	Apstra Solution
M	What is Intent Based Networking
N	REST API Guide
O	AOS Service Oriented Dashboards
P	IP Version 4 (IPv4) Datagram General Format
Q	A Primer to Intent Driven Networking